F2-04RTD
4-Channel RTD Input

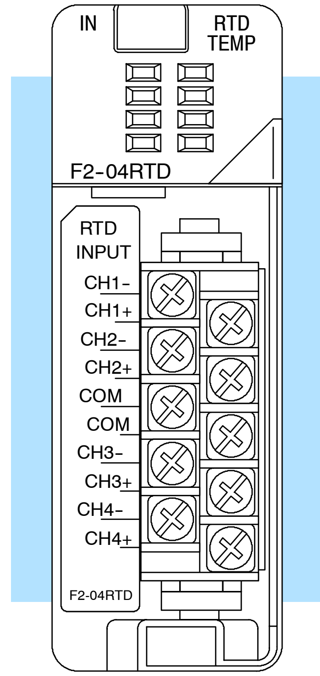
6

In This Chapter. . . .

Module Specifications

The F2-04RTD 4-Channel Resistive Temperature Detector Input Module provides several features and benefits:

- Provides four RTD input channels with 0.1°F resolution.
- Automatically converts type Pt100Ω, jPt100Ω, Pt1000Ω, Cu 25Ω, Cu10Ω signals into direct temperature readings. No extra scaling or complex conversion is required.
- Temperature data format is selectable between ° F or ° C , magnitude plus sign, or 2's complement.
- Precision lead wire resistance compensation by dual matched current sources and ratiometric measurements.
- Temperature calculation and linearization are based on data provided by the National Institute of Standards and Technology (NIST).
- Diagnostics features include detection of short circuits and input power disconnection.



Module Calibration

The module automatically re-calibrates every five seconds to remove any offset and gain errors. The F2-04RTD module requires no user calibration. However, if your process requires calibration, it is possible to correct the RTD tolerance using ladder logic. You can subtract or add a constant to the actual reading for that particular RTD.

RTD Input Configuration Requirements

The F2-04RTD module requires 32 discrete input points from the CPU. The module can be installed in any slot of a DL205 system, including remote bases. The limiting factors on the number of analog modules used are:

- For local and local expansion systems, the available power budget and number of discrete I/O points.
- For remote I/O systems, the available power budget and number of remote I/O points.

Check the user manual for your particular CPU model for more information regarding the available power budget and number of local, local expansion or remote I/O points.

NOTE: DL230 CPUs with firmware release version 1.6 or later, DL240 CPUs with firmware release 2.5 or later, DL250 CPUs with firmware release version 1.06 or later are required for proper operation.

The following table provides the specifications for the F2-04RTD module. Review these specifications to make sure the module meets your application requirements.

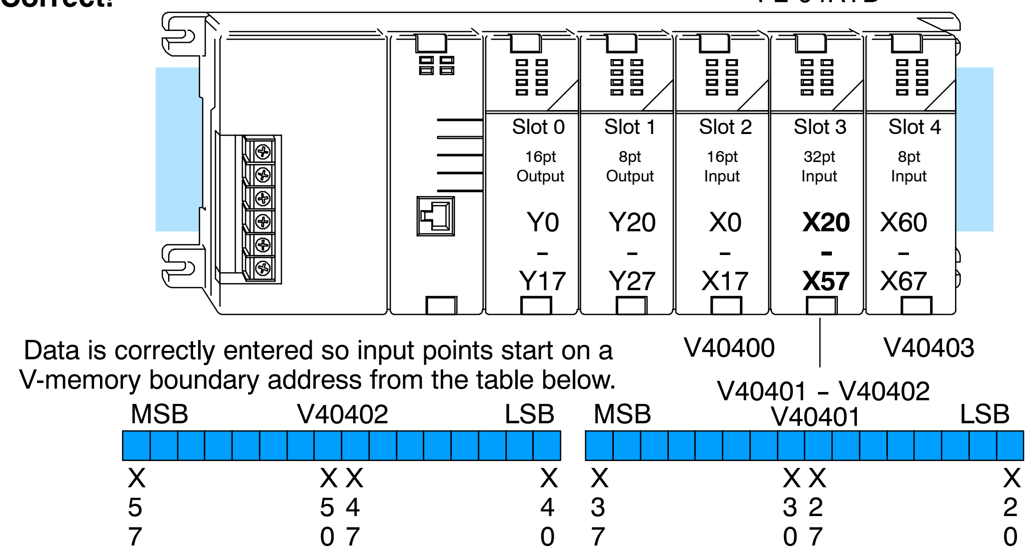
Input Specifications

Number of Channels	4, differential inputs
Input Ranges	Pt100 -200°C to 850°C (-328°F to 1562°F) Pt 1000 -200°C to 595°C (-328°F to 1103°F) jPt100 -38°C to 450°C (-36°F to 842°F) 10ΩCu. -200°C to 260°C (-328°F to 500°F) 25ΩCu. -200°C to 260°C (-328°F to 500°F)
Resolution	±0.1 °C, ±0.1 °F (± 3276.7)
Absolute Maximum Ratings	Fault protected input, ±50 Vdc
Converter Type	Charge balancing, 24-bit
Sampling Rate	160 msec per channel
Linearity Error (End to End)	±0.05 °C maximum, ±0.01 °C typical
PLC Update Rate	4 Channels/Scan max. 240/250-1/260 CPU 1 Channel/Scan max. 230 CPU
Temperature Drift	5ppm per °C (maximum)
Maximum Inaccuracy	±1°C
RTD Excitation Current	200 μA
Common Mode Range	0-5 VDC
Notch Filter	>100dB notches @ 50/60 Hz f_{-3dB} = 13.1 Hz
Digital Input Points Required	32 (X) input points 15 binary data bits, 1 sign bit, 2 channel ID bits 4 fault bits
Power Budget Requirement	90 mA @ 5 VDC (from base)
Operating Temperature	0° to 60° C (32° to 140° F)
Storage Temperature	-20° to 70° C (-4° to 158° F)
Relative Humidity	5 to 95% (non-condensing)
Environmental air	No corrosive gases permitted
Vibration	MIL STD 810C 514.2
Shock	MIL STD 810C 516.2
Noise Immunity	NEMA ICS3-304

Special Placement Requirements (DL230 and Remote I/O Bases)

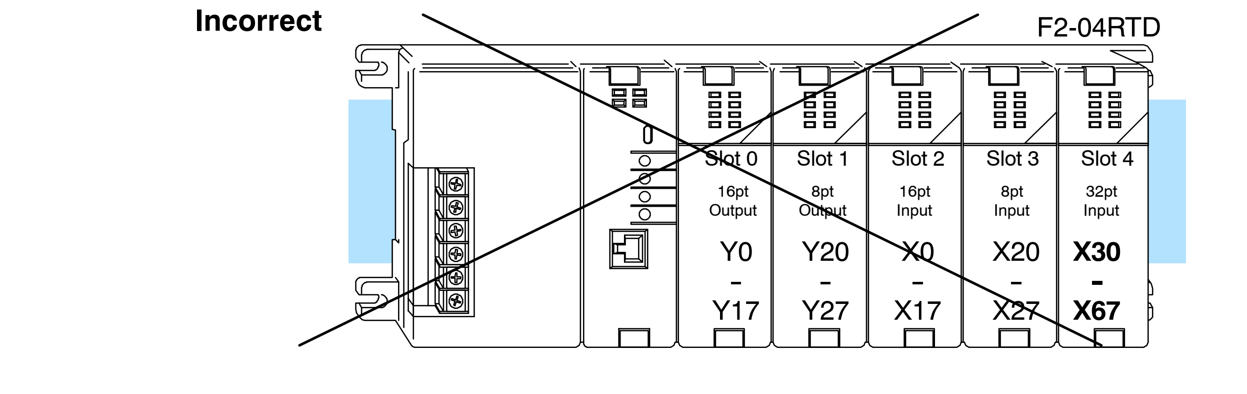
It is important to examine the configuration if you are using a DL230 CPU. As you can see in the section on writing the program, you use V-memory locations to send the analog data. If you place the module so that the input points do not start on a V-memory boundary, the instructions cannot access the data. This also applies when placing this module in a remote base using a D2–RSS in the CPU slot. See the table below.

Correct!



Data is correctly entered so input points start on a V-memory boundary address from the table below.

Incorrect



Data is split over three locations, so instructions cannot access data from a DL230.

To use the V-memory references required for a DL230 CPU, the *first* input address assigned to the module must be one of the following X locations. The table also shows the V-memory addresses that correspond to these X locations.

X	X0	X20	X40	X60	X100	X120	X140	X160
V	V40400	V40401	V40402	V40403	V40404	V40405	V40406	V40407

Setting the Module Jumpers

Jumper Locations

Locate the bank of seven jumpers (J8) on the PC board. Notice that the description of each jumper is on the PC board. You can select the following options by installing or removing the jumpers:

- Number of channels: 1 thru 4.
- The input type: 10 Ω (ohms) or 25 Ω copper RTDs; jPt 100 Ω, Pt 100 Ω or Pt 1000 Ω RTDs
- Temperature conversion: 2's complement or magnitude plus sign format in Fahrenheit or Celsius.

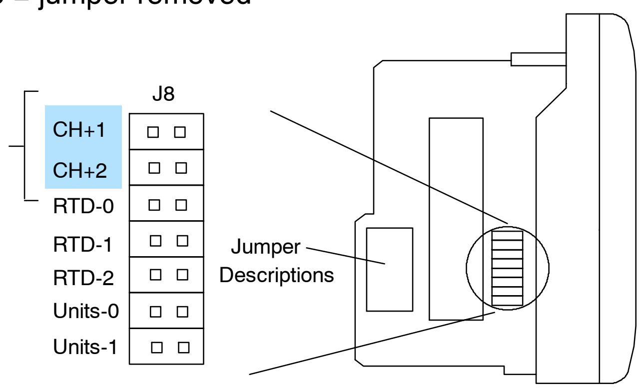
To prevent losing a jumper when it is removed, store it near its original location by sliding one of its sockets over a single pin.

Selecting the Number of Channels

The two jumpers labeled **CH+1** and **CH+2** are used to select the number of channels that will be used. The factory default setting is four-channel operation (both jumpers installed). Any unused channels are not processed. For example, if you select channels 1 thru 3, channel 4 will be inactive. The table shows how to arrange the jumpers to select the number of channels.

X = jumper installed, empty space = jumper removed

Number of Channels	Jumper	
	CH+1	**CH+2**
1		
2	X	
3		X
4	X	X



Setting Input Type

The jumpers labeled **RTD-0, RTD-1,** and **RTD-2** are used to select the type of RTD. The module can be used with many types of RTDs. All channels of the module must be the same RTD type.

The default setting from the factory is Pt100 Ω (RTD-2 comes with the jumper removed). This selects the DIN 43760 European type RTD. European curve type RTDs are calibrated to DIN 43760, BS1905, or IEC751 specifications which is .00385 Ω /Ω / ° C (100° C = 138.5Ω).

The jPt100 Ω type is used for the American curve (.00392 Ω/Ω/° C), platinum 100 Ω RTDs. The 10 Ω and 25 Ω RTD settings are used with copper RTDs.

The table shows how to arrange the jumpers to set the input type.

X = jumper installed, empty space = jumper removed

RTD Inputs	Jumper Pins		
	RTD-0	**RTD-1**	**RTD-2**
Cu 10 Ω			
Cu 25 Ω	X		
jPt100 Ω		X	
Pt100 Ω	X	X	
Pt1000 Ω			X

Selecting the Conversion Units

Use the last two jumpers, **Units-0** and **Unit-1**, to set the conversion unit. The options are magnitude + sign or 2's complement in Fahrenheit or Celsius. The module comes from the factory with both jumpers installed for magnitude + sign conversion in Fahrenheit.

All RTD types are converted into a direct temperature reading in either Fahrenheit or Celsius. The data contains one implied decimal place. For example, a value in V-memory of 1002 would be 100.2°C or °F.

Negative temperatures can be represented in either 2's complement or magnitude plus sign form. If the temperature is negative, the most significant bit in the V-memory location is set (X17).

The 2's complement data format may be required to correctly display bipolar data on some operator interfaces. This data format could also be used to simplify averaging a bipolar signal. To view this data format in ***Direct***Soft32, select Signed Decimal.

The table shows how to arrange the jumpers.

X = jumper installed, empty space = jumper removed.

Jumper	Temperature Conversion Units			
	Magnitude + Sign		2's Complement	
	°F	°C	°F	°C
Units-0	X		X	
Units-1	X	X		

Connecting the Field Wiring

Wiring Guidelines

Your company may have guidelines for wiring and cable installation. If so, you should check those before you begin the installation. Here are some general things to consider:

- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the transmitter source. *Do not* ground the shield at both the module and the source.
- Do not run the signal wiring next to large motors, high current switches, or transformers. This may cause noise problems.
- Route the wiring through an approved cable housing to minimize the risk of accidental damage. Check local and national codes to choose the correct method for your application.

RTD – Resistance Temperature Detector

Use shielded RTDs whenever possible to minimize noise on the input signal. Ground the shield wire at one end only. Connect the shield wire to the COM terminal.

Lead Configuration for RTD Sensors

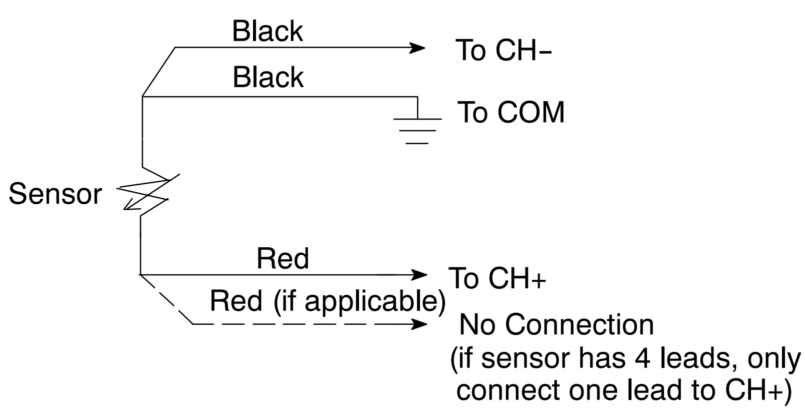
The suggested three-lead configuration shown below provides one lead to the CH+ terminal, one lead to the CH- terminal, and one lead to the common terminal. Compensation circuitry nulls out the lead length for accurate temperature measurements.

Some sensors have four leads. When making connections, do not connect the second lead to the CH+ input; leave that lead unconnected.

Do not use configurations having only one lead connected to each input. There is no compensation and temperature readings will be inaccurate.

This module has low RTD excitation current, worst-case dissipation is only .016 mW.

Wiring Connections For Typical RTD Sensor



Ambient Variations in Temperature

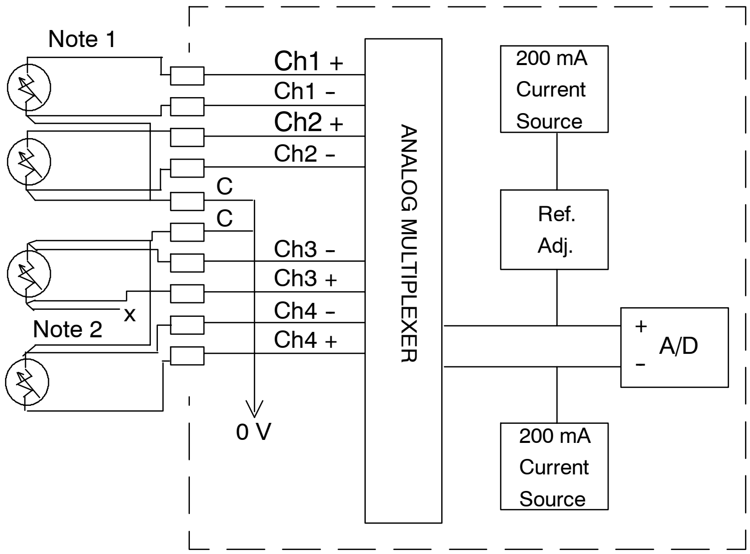
The F2-04RTD module has been designed to operate within the ambient temperature range of 0°C to 60°C.

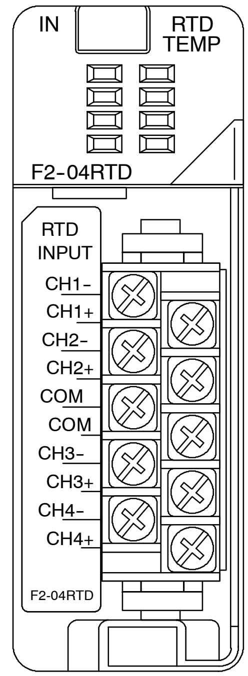
Precision analog measurement with no long term temperature drift is assured by a chopper stabilized programmable gain amplifier, ratiometric referencing, and automatic offset and gain calibration.

Wiring Diagram

The F2-04RTD module has a removable connector to make wiring easier. Simply squeeze the top and bottom retaining clips and gently pull the connector from the module.

Wiring Diagram





Notes:

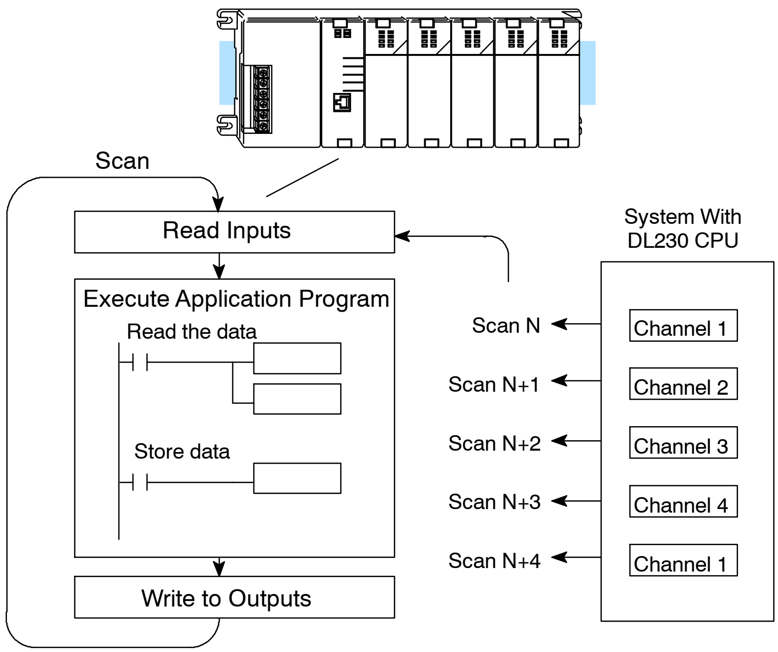
1. The three wires connecting the RTD to the module must be the same type and length. Do not use the shield or drain wire for the third connection.
2. If the RTD sensor has four wires, the plus (+) sense wire should be left unconnected as shown.

Module Operation

Before you begin writing the control program, it is important to take a few minutes to understand how the module processes and represents the analog signals.

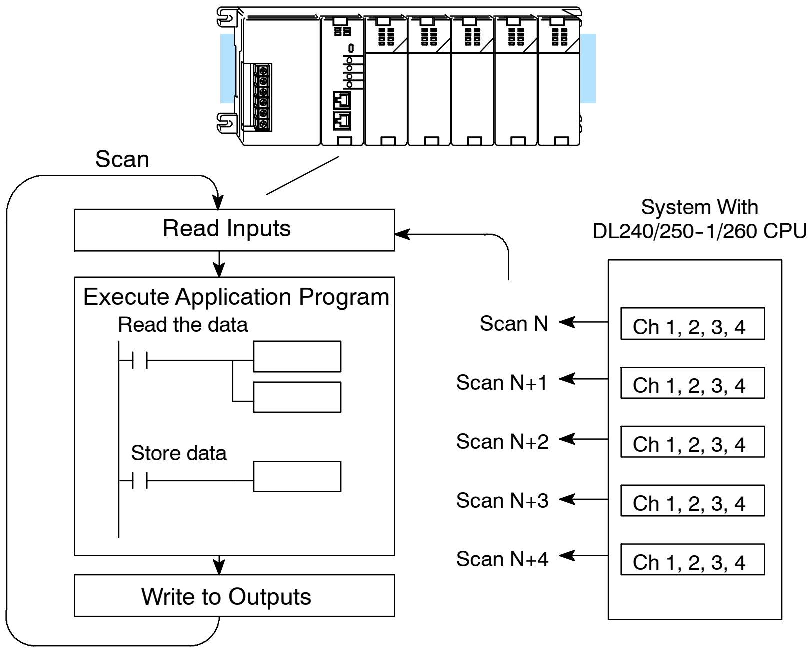
Channel Scanning Sequence for a DL230 CPU (Multiplexing)

The F2-04RTD module can supply different amounts of data per scan, depending on the type of CPU you are using. The DL230 can obtain one channel of data per CPU scan. Since there are four channels, it can take up to four scans to get data for all channels. Once all channels have been scanned the process starts over with channel 1. Unused channels are not processed, so if you select only two channels, each channel will be updated every other scan. The multplexing method can also be used for the DL240/250–1/260 CPUs.



Channel Scanning Sequence for a DL240, DL250–1 or DL260 CPU (Pointer Method)

If you are using a DL240, DL250–1 or DL260 CPU, you can obtain all four channels of input data in one scan. This is because the DL240/250–1/260 CPUs support special V-memory locations that are used to manage the data transfer. This is discussed in more detail in the section on Writing the Control Program.



Analog Module Updates

Even though the channel updates to the CPU are synchronous with the CPU scan, the module asynchronously monitors the analog transmitter signal and converts the signal to a 16-bit binary representation. This enables the module to continuously provide accurate measurements without slowing down the discrete control logic in the RLL program.

The time required to sense the temperature and copy the value to V-memory is 160 milliseconds minimum to 640 milliseconds plus 1 scan time maximum (number of channels x 160 msec + 1 scan time).

Writing the Control Program

Reading Values: Pointer Method and Multiplexing

There are two methods of reading values:

- The pointer method
- Multiplexing

You *must* use the multiplexing method when using a DL230 CPU. You must also use the multiplexing method with remote I/O modules (the pointer method will not work). You can use either method when using DL240, DL250-1 and DL260 CPUs, but for ease of programming it is strongly recommended that you use the pointer method.

Pointer Method

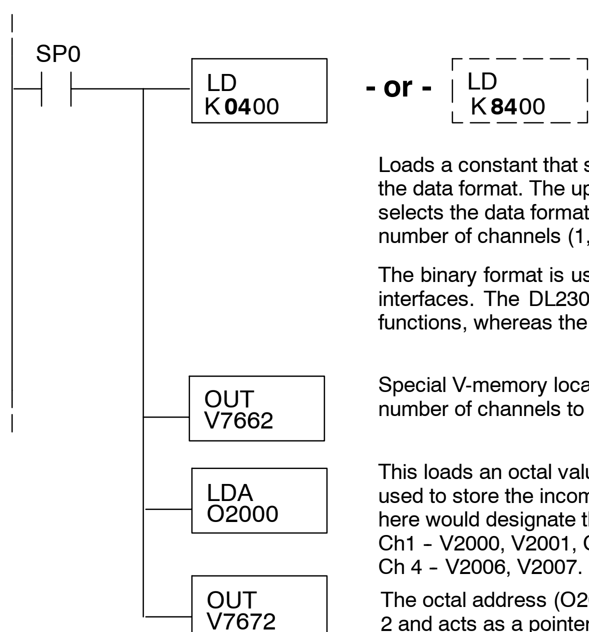
230	240	250-1	260
✗	✓	✓	✓

The CPU has special V-memory locations assigned to each base slot that greatly simplify the programming requirements. These V-memory locations:

- specify the number of channels to scan.
- specify the storage locations.

The example program shows how to setup these locations. Place this rung anywhere in the ladder program, or in the initial stage if you are using stage programming instructions. This is all that is required to read the data into V-memory locations. Once the data is in V-memory, you can perform math on the data, compare the data against preset values, and so forth. V2000 is used in the example, but you can use any user V-memory location. In the examples, the module is installed in slot 2. You should use the V-memory locations used in your application. The pointer method automatically converts values to BCD.

NOTE: DL240 CPUs with firmware release version 2.5 or later and DL250 CPUs with firmware release version 1.06 or later support this method. Use the DL230 multiplexing example if your firmware revision is earlier (verify that the addresses in the CPU are zero).



Loads a constant that specifies the number of channels to scan and the data format. The upper byte, most significant nibble (MSN) selects the data format (0=BCD, 8=Binary), the LSN selects the number of channels (1, 2, 3, or 4).

The binary format is used for displaying data on some operator interfaces. The DL230/240 CPUs do not support binary math functions, whereas the DL250 does.

Special V-memory location assigned to slot 2 that contains the number of channels to scan.

This loads an octal value for the first V-memory location that will be used to store the incoming data. For example, the O2000 entered here would designate the following addresses:
Ch1 - V2000, V2001, Ch 2 - V2002, V2003, Ch 3 - V2004, V2005, Ch 4 - V2006, V2007.

The octal address (O2000) is stored here. V7672 is assigned to slot 2 and acts as a pointer, which means the CPU will use the octal value in this location to determine exactly where to store the incoming data.

The tables below show the special V-memory locations used by the DL240, DL250-1 and DL260 for the CPU base and local expansion base I/O slots. Slot 0 (zero) is the module next to the CPU or D2-CM module. Slot 1 is the module two places from the CPU or D2-CM, and so on. Remember, the CPU only examines the pointer values at these locations after a mode transition. Also, if you use the DL230 (multiplexing) method, verify that these addresses in the CPU are zero.

The Table below applies to the DL240, DL250-1 and DL260 CPU base.

CPU Base: Analog Input Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V7660	V7661	V7662	V7663	V7664	V7665	V7666	V7667
Storage Pointer	V7670	V7671	V7672	V7673	V7674	V7675	V7676	V7677

The Table below applies to the DL250-1 or DL260 expansion base 1.

Expansion Base D2-CM #1: Analog Input Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36000	V36001	V36002	V36003	V36004	V36005	V36006	V36007
Storage Pointer	V36010	V36011	V36012	V36013	V36014	V36015	V36016	V36017

The Table below applies to the DL250-1 or DL260 expansion base 2.

Expansion Base D2-CM #2: Analog Input Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36100	V36101	V36102	V36103	V36104	V36105	V36106	V36107
Storage Pointer	V36110	V36111	V36112	V36113	V36114	V36115	V36116	V36117

The Table below applies to the DL260 CPU expansion base 3.

Expansion Base D2-CM #3: Analog Input Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36200	V36201	V36202	V36203	V36204	V36205	V36206	V36207
Storage Pointer	V36210	V36211	V36212	V36213	V36214	V36215	V36216	V36217

The Table below applies to the DL260 CPU expansion base 4.

Expansion Base D2-CM #4: Analog Input Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36300	V36301	V36302	V36303	V36304	V36305	V36306	V36307
Storage Pointer	V36310	V36311	V36312	V36313	V36314	V36315	V36316	V36317

Negative Temperature Readings with Magnitude Plus Sign (Pointer Method)

230	240	250-1	260
×	✓	✓	✓

With bipolar ranges, you need some additional logic to determine whether the value being returned represents a positive voltage or a negative voltage. For example, you may need to know the direction for a motor. There is a simple solution:

- If you are using bipolar ranges and you get a value greater than or equal to 8000_H, the value is negative.
- If you get a value less than or equal to $7FFF_H$, the value is positive.

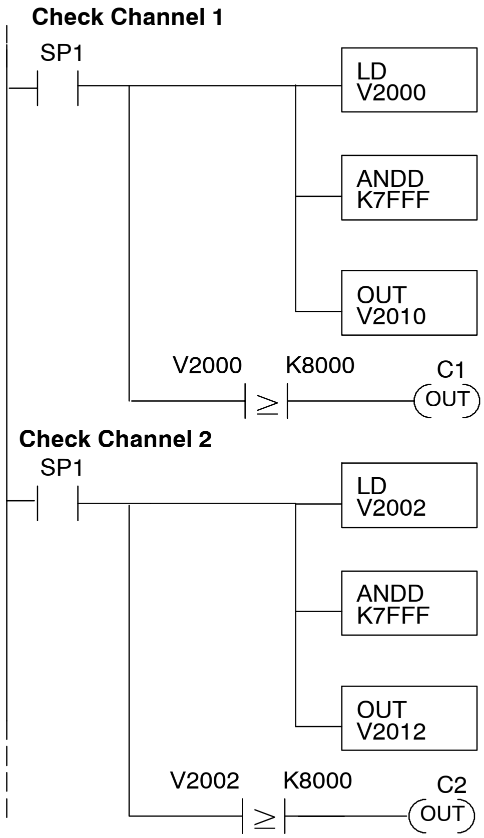
The sign bit is the most significant bit, which combines 8000_H to the data value. If the value is greater than or equal to 8000_H, you only have to mask the most significant bit and the active channel bits to determine the actual data value.

NOTE: DL240 CPUs with firmware release version 2.5 or later and DL250 CPUs with firmware release version 1.06 or later support this method. Use the DL230 multiplexing example if your firmware revision is earlier.

The following two programs show how you can accomplish this. The first example uses magnitude plus sign (binary) and the second example uses magnitude plus sign (BCD).

Since you always want to know when a value is negative, these rungs should be placed *before* any other operations that use the data, such as math instructions, scaling operations, and so forth. Also, if you are using stage programming instructions, these rungs should be in a stage that is always active. Note: you only need this logic for each channel that is using bipolar input signals. The following examples only show two channels.

Magnitude Plus Sign (Binary)



Load channel 1 data from V-memory into the accumulator. Contact SP1 is always on.

This instruction masks the sign bit of the binary data, if it is set. Without this step, negative values will not be correct so do not forget to include it.

Put the actual signal value in V2010. Now you can use the data normally.

Channel 1 data is negative when C1 is on (a value of -1.0 reads as 8010, -2.0 is 8020, etc.).

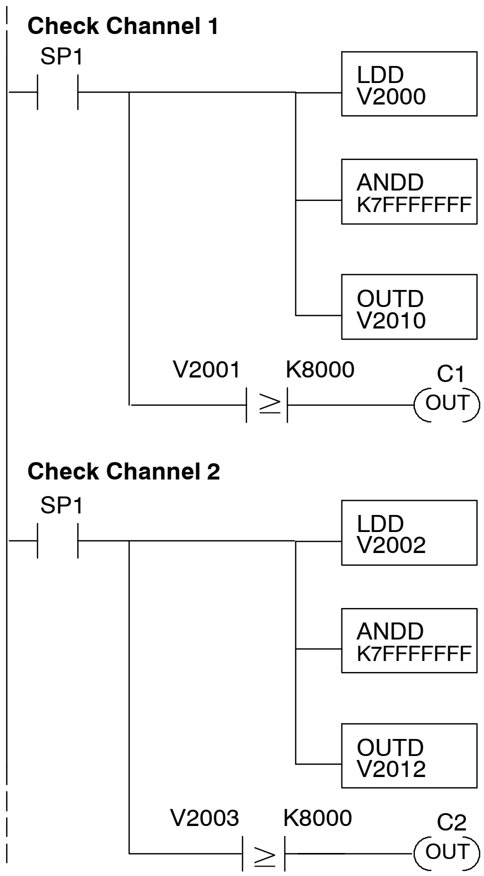
Load channel 2 from V-memory into the accumulator. Contact SP1 is always on.

This instruction masks the sign bit of the binary data, if it is set. Without this step, negative values will not be correct so do not forget to include it.

Put the actual signal value in V2012. Now you can use the data normally.

Channel 2 data is negative when C2 is on (a value of -1.0 reads as 8010, -2.0 is 8020, etc.).

Magnitude Plus Sign (BCD)



Load channel 1 data from V-memory into the accumulator. Remember, the data can be negative. Contact SP1 is always on.

This instruction masks the sign bit of the BCD data, if it is set. Without this step, negative values will not be correct so do not forget to include it.

Put the actual signal value in V2010. Now you can use the data normally.

Channel 1 data is negative when C1 is on (a value of -1.0 reads as 8000 0010, -2.0 is 8000 0020, etc.).

Load channel 2 from V-memory into the accumulator. Remember, the data can be negative. Contact SP1 is always on.

This instruction masks the sign bit of the BCD data, if it is set. Without this step, negative values will not be correct so do not forget to include it.

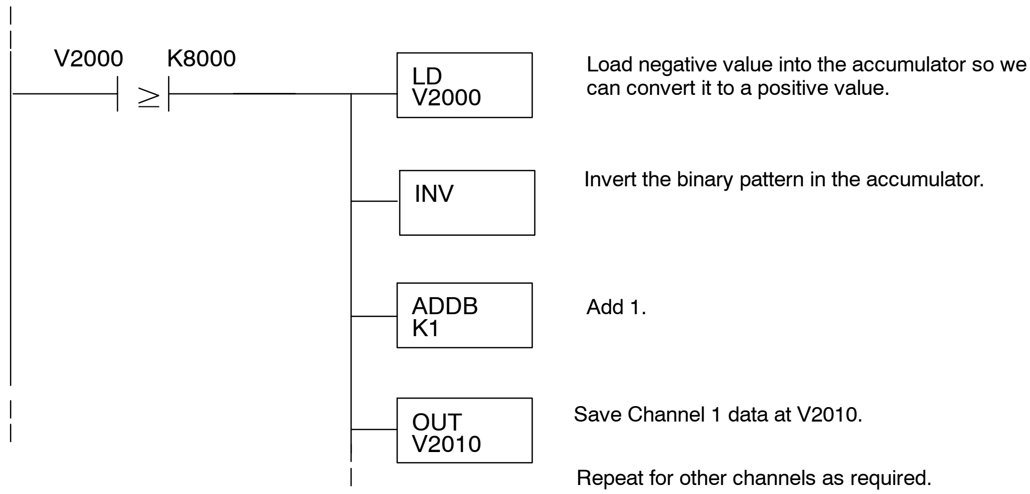
Put the actual signal value in V2012. Now you can use the data normally.

Channel 2 data is negative when C2 is on (a value of -1.0 reads as 8000 0010, -2.0 is 8000 0020, etc.).

Negative Temperatures 2's Complement (Binary / Pointer Method)

230	240	250-1	260
✗	✓	✓	✓

You can use the 2's complement mode for negative temperature display purposes, while at the same time using the magnitude plus sign of the temperature in your control program. The ***Direct***SOFT32 element Signed Decimal is used to display negative numbers in 2's complement form. To find the absolute value of a negative number in 2's complement, invert the number and add 1 as shown in the following example:



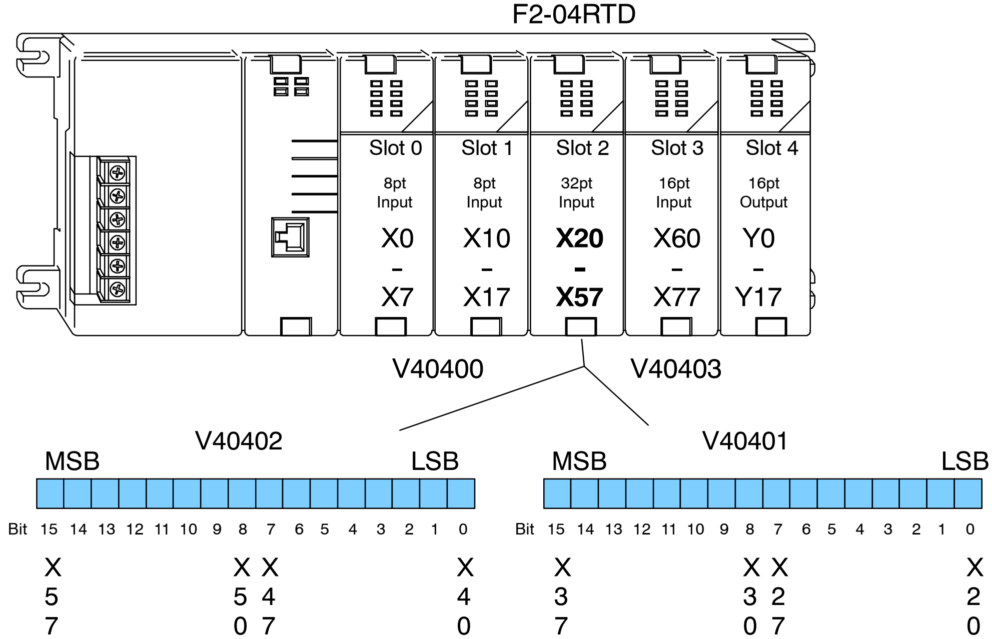
Understanding the Input Assignments (Multiplexing Ladder Only)

230	240	250-1	260
✓	✓	✓	✓

You may recall that this module appears to the CPU as a 32-point discrete input module. You can use these points to obtain:

- An indication of which channel is active
- The digital representation of the analog signal
- Module diagnostic information

Since all input points are automatically mapped into V-memory, it is very easy to determine the location of the data word that will be assigned to the module.



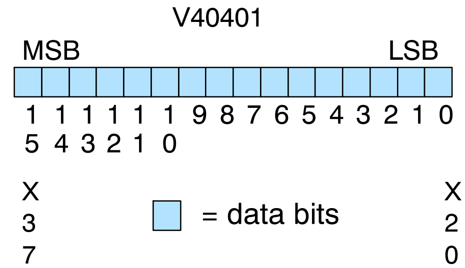
Remember, when using DL230 CPUs input points must start on a V-memory boundary. To use the V-memory references required for a DL230 CPU, the *first* input address assigned to the module must be one of the following X locations. The table also shows the V-memory addresses that correspond to these X locations.

X	X0	X20	X40	X60	X100	X120	X140	X160
V	V40400	V40401	V40402	V40403	V40404	V40405	V40406	V40407

Analog Data Bits

The first 16 bits represent the analog data in binary format.

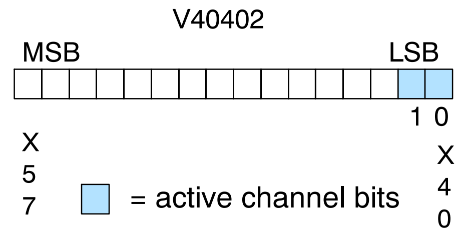
Bit	Value	Bit	Value
0	1	8	256
1	2	9	512
2	4	10	1024
3	8	11	2048
4	16	12	4096
5	32	13	8192
6	64	14	16384
7	128	15	32768



Active Channel Bits

The active channel bits represent the multiplexed channel selections in binary format.

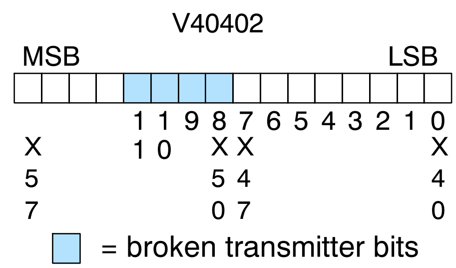
Bit 1	Bit 0	Channel
0	0	1
0	1	2
1	0	3
1	1	4



Broken Transmitter Bits (Pointer and Multiplexing Ladder Methods)

The broken transmitter bits are on when the corresponding RTD is open.

Bit	Channel
8	1
9	2
10	3
11	4

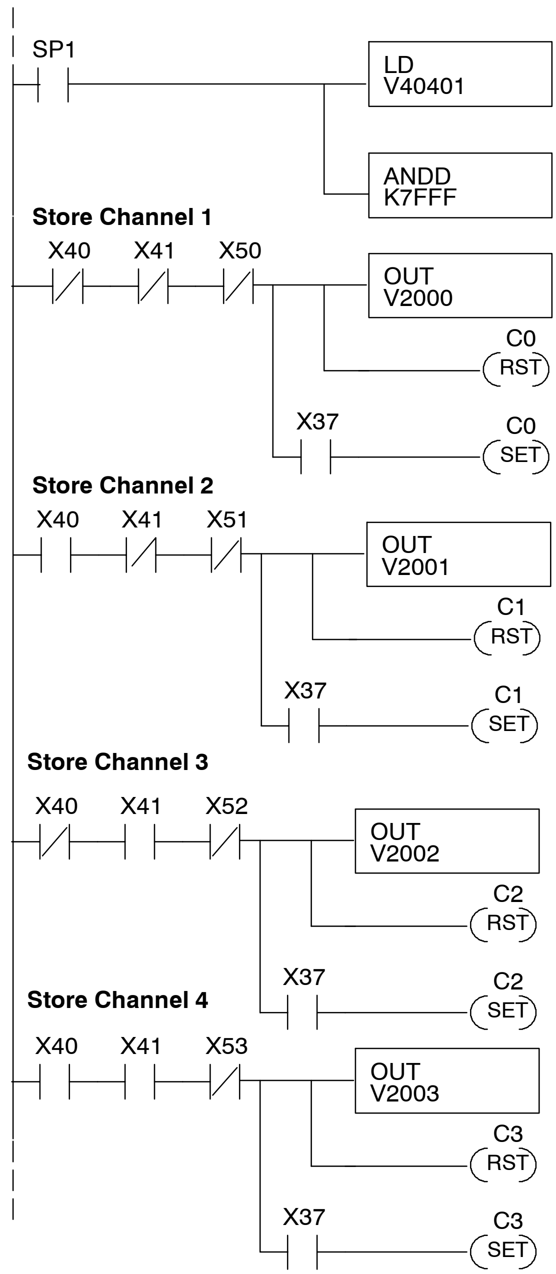


Reading Magnitude Plus Sign Values (Multiplexing)

✓ 230 ✓ 240 ✓ 250–1 ✓ 260

The DL230 CPU *does not* have the special V-memory locations that allow you to automatically enable the data transfer. Since all channels are multiplexed into a single data word, the control program must be setup to determine which channel is being read. Since the module appears as X input points to the CPU, it is very easy to use the active channel status bits to determine which channel is being monitored.

NOTE: DL230 CPUs with firmware release version 1.6 or later required for multiplexing ladder.



Loads the complete data word into the accumulator. The V-memory location depends on the I/O configuration. See Appendix A for the memory map.

This instruction masks the sign bit. Without this, the values used will not be correct so do not forget to include it.

When X40, X41, and X50 are off, channel 1 data is stored in V2000. C0 is reset to indicate that channel 1's value is positive.

If X37 is on, the data value represents a negative temperature. C0 is set to indicate that channel 1's value is negative.

When X40 is on and X41 and X51 are off, channel 2 data is stored in V2001. C1 is reset to indicate that channel 2's value is positive.

If X37 is on, the data value represents a negative temperature. C1 is set to indicate that channel 2's value is negative.

When X40 and X52 are off and X41 is on, channel 3 data is stored in V2002. C2 is reset to indicate that channel 3's value is positive.

If X37 is on, then the data value represents a negative temperature. C2 is set to indicate that channel 3's value is negative.

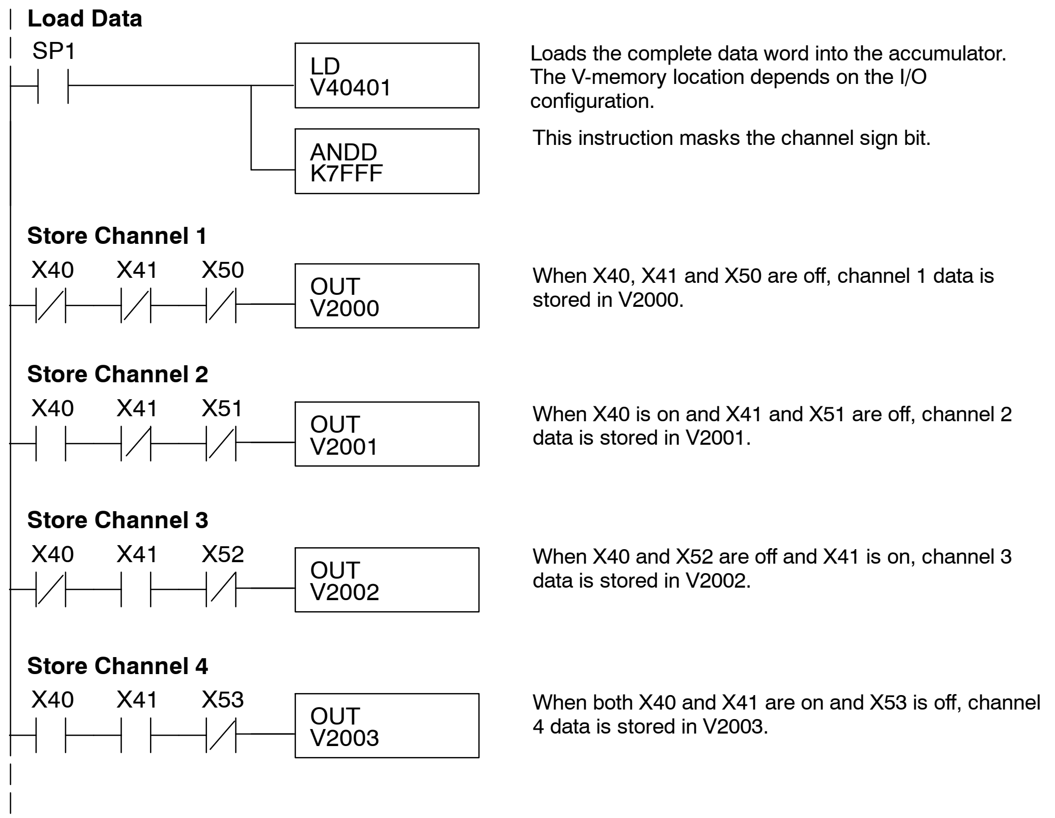
When both X40 and X41 are on and X53 is off, channel 4 data is stored in V2003. C3 is reset to indicate that channel 4's value is positive.

If X37 is on, the data value represents a negative temperature. C3 is set to indicate that channel 4's value is negative.

Reading 2's Complement Values (Multiplexing)

230	240	250–1	260
✓	✓	✓	✓

The DL230 CPU *does not* have the special V-memory locations that allow you to automatically enable the data transfer. Since all channels are multiplexed into a single data word, the control program must be setup to determine which channel is being read. Since the module appears as X input points to the CPU, it is very easy to use the active channel status bits to determine which channel is being monitored. The 2's complement data format may be required to correctly display bipolar data on some operator interfaces. This data format could also be used to simplify averaging a bipolar signal. To view this data format in ***Direct***SOFT32, select Signed Decimal.



Scaling the Input Data

No scaling of the input temperature is required. The readings directly reflect the actual temperatures. For example: a reading of 8482 is 848.2 °C, a reading of 16386 is –0.2°C. (magnitude plus sign) and a reading of 32770 is –0.2°C (2's complement).

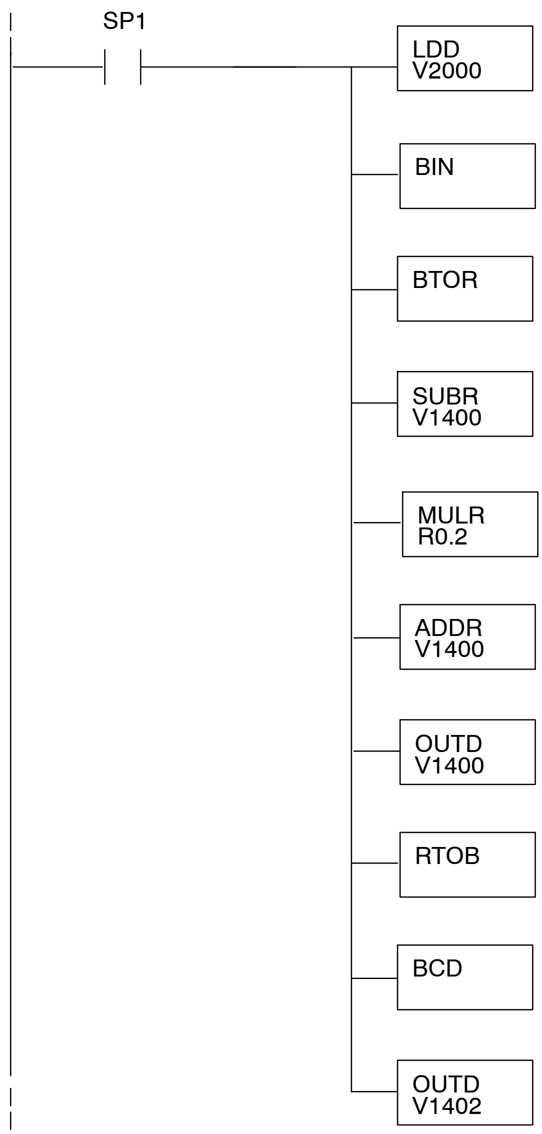
Filtering Input Noise (DL250-1, DL260 CPUs Only)

230	240	250-1	260
✕	✕	✓	✓

Add the following logic to filter and smooth analog input noise in DL250-1 and DL260 CPUs. This is especially useful when using PID loops. Noise can be generated by the field device and/or induced by field wiring.

The analog value in BCD is first converted to a binary number because there is not a BCD-to-real conversion instruction. Memory location V1400 is the designated workspace in this example. The MULR instruction is the filter factor, which can be from 0.1 to 0.9. The example uses 0.2. A smaller filter factor increases filtering. You can use a higher precision value, but it is not generally needed. The filtered value is then converted back to binary and then to BCD. The filtered value is stored in location V1402 for use in your application or PID loop.

NOTE: Be careful not to do a multiple number conversion on a value. For example, if you are using the pointer method to get the analog value, it is in BCD and must be converted to binary. However, if you are using the conventional method of reading analog and are masking the first fifteen bits, then it is already in binary and no conversion using the BIN instruction is needed. Also, if you are using the conventional method, change the LLD V2000 instruction to LD V2000.



LDD V2000: Loads the analog signal, which is a BCD value and has been loaded from V-memory location V2000, into the accumulator. Contact SP1 is always on.

BIN: Converts the BCD value in the accumulator to binary. Remember, this instruction is not needed if the analog value is originally brought in as a binary number.

BTOR: Converts the binary value in the accumulator to a real number.

SUBR V1400: Subtracts the real number stored in location V1400 from the real number in the accumulator, and stores the result in the accumulator. V1400 is the designated workspace in this example.

MULR R0.2: Multiplies the real number in the accumulator by 0.2 (the filter factor), and stores the result in the accumulator. This is the filtered value.

ADDR V1400: Adds the real number stored in location V1400 to the real number filtered value in the accumulator, and stores the result in the accumulator.

OUTD V1400: Copies the value in the accumulator to location V1400.

RTOB: Converts the real number in the accumulator to a binary value, and stores the result in the accumulator.

BCD: Converts the binary value in the accumulator to a BCD number. **Note:** The BCD instruction is not needed for PID loop PV (loop PV is a binary number).

OUTD V1402: Loads the BCD number filtered value from the accumulator into location V1402 to use in your application or PID loop.